REMARKS

Applicant respectfully requests favorable reconsideration of this application.

Telephone Interview

Applicant respectfully thanks the Examiner for her kindness and courtesy in conducting a telephone interview with Applicant's undersigned representative. The gist of the interview was to determine whether (1) the Examiner believed, like Applicant, that the Anders reference discloses a very different invention than the present invention, but that the Examiner simply had a different interpretation of Applicant's claims than Applicant or (2) the Examiner had a similar interpretation of the claims as Applicant, but had a different understanding of Anders than Applicant, i.e., the Examiner believes that Anders discloses an invention similar to Applicant's invention.

Some progress was made on this front and it appears that the dispute may revolve around differing understandings of the claims and Applicant will proceed with this assumption.

During the interview, the Examiner indicated that she appreciated Applicant's concerns and recommended that Applicant prepare a response to the final Office Action and indicated that she would fully consider any such response.

Background

This case had previously been appealed and an appeal brief was filed September 16, 2005. The previous Office Action reopened prosecution and asserted a new ground of rejection. The present final Office Action maintains those grounds for rejection.

Applicant respectfully traverses.

The Present Invention

The present invention is a method and apparatus for loading web pages, including supplemental files such as pictures, sound files, video files, etc., at a browser.

One of the problems of the prior art is that browsers read the HTML code in a Web page from left to right and from top to bottom. Accordingly, the browser encounters the embedded references to supplemental files in the order in which they are encountered while reading the page. The browser will send requests back to the server for those supplemental files in the order that the browser encounters the references while reading the HTML code. Since a browser has a limited number of ports, the supplemental files may not be retrieved and loaded in the most efficient manner. For instance, if a browser has 4 ports and the requested page has 15 supplemental files, in which the first 4 referenced supplemental files are large files and the next 11 are small files, the browser may take a long time to download the first 4 large files, while the person sitting at the client browser watches a largely or completely

blank screen. If the 10 small files could be downloaded first, the browsing experience for the person can be much improved because he/she could then have something to look at while waiting for the 4 large files to download.

The present invention addresses this concern without the need to modify the browser software in any way. In accordance with the invention, the order in which supplemental files referenced in a Web page are downloaded from the server to the requesting client is specified by the designer of the HTML code of the Web page and controlled at the server side regardless of the order in which the client-side Web browser encounters and requests the supplemental files. Particularly, each supplemental file referenced in a Web page has a sequence number associated with it. In a preferred embodiment, the sequence number is provided as an additional attribute of the tag that calls the supplemental file. Since the client Web browser is a standard Web browser, it will have no idea what the sequence attribute is, which is acceptable since browsers generally will ignore any attribute within a tag that it does not understand. However, at the server-side, when the page is requested, the server parses the page before sending it to the requesting client to find the tags for the supplemental files embedded within the page and reads the associated sequence number attributes. It then builds a queue for serving the supplemental files to the client machine, the supplemental files being queued in the order dictated by the sequence numbers.

Thereafter, regardless of the order in which the browser returns requests for the supplemental files, the server will serve the supplemental files in the order dictated by the queue. Existing browsers already are equipped to receive and cache files and associate such cached files with files referenced in an HTML page. Accordingly, the fact that the supplemental files referenced in a Web page may be received in an order different from the order in which the browser requests them is of no consequence. Specification, page 6, lines 8–13. Accordingly, the invention resides entirely at the server side and will work with any Web browser.

The Anders Reference

Anders discloses a method and apparatus for serving Web pages, including supplemental files, to a requesting client. However, the method and apparatus disclosed in Anders is entirely different than that of the present invention. Unlike the present invention, Anders requires the Web browser software to be modified to function with the invention. See col. 8, lines 2-6 (which describes the need for a Jammer unpacker "on the client"), col. 8, lines 51-54, and col. 12, line 65 – col. 14, line 7 (which describes in detail the software needed at the browser to implement the invention).

Anders' scheme is entirely different than Applicant's. With reference to Anders' Figure 8, the server transmits the requested page to the requesting client in a particular data stream format 190 that includes the data for the main object (the Web page) and the data for the supplemental objects (such as embedded pictures, etc.) in data entries

(packets, such as packets 181-189) that are interleaved with each other in an order selected by the developer. More particularly, the data stream 190 comprises a stream header 180 at the beginning of the stream followed by data definition entries and HTML data entries. Each data definition entry, e.g., 181, 182, 185, 187, defines a supplemental object/file present in the Web page data stream. There is one data definition entry per object/file. The HTML data entries are the actual data of the objects/files (including the main file as well as the supplemental files). Each file will typically consist of many HTML data entries that the browser assembles together to render the whole file. The data definition entry that defines any given object/file must precede the first HTML data entry of that file in order for the browser to know what to do with those HTML data entries when it receives them. Col. 7, line 57 - col. 8, line 36.

The basic premise of Anders' invention is that the publisher 210 (Fig. 11) interleaves the data for the entire web page in a way dictated by itself and serves it to the client that way. The browser, upon receiving each data definition entry, creates an entry in an unpacked object cache (UOC). Then, when the browser starts receiving the HTML data entries corresponding to the supplemental file identified by any given data definition entry, it will append that HTML data to the entry it created in its UOC. In Anders, the browser receives the tags identifying the supplemental files in the order dictated by the data stream 190. Accordingly, the browser may be receiving data of a supplemental file before it receives the HTML data entry that contains the reference to that supplemental file. That is not a problem. Particularly, when the browser reaches

the reference to the supplemental file that it has already started downloading and caching in its UOC, the UOC simply forwards the cached data to the browser for rendering.

The Krishnan Reference

Krishnan is the secondary reference in the rejections of claims 1-22 and has been cited for a very narrow proposition, namely, teaching "indicating an order". The Office has referred only to col. 3, line 45 to col. 4, line 9, which reads:

> In FIG. 1, a WEB browser application 101 is shown executing on a client computer system 102, which communicates with a server computer system 103 by sending and receiving HTTP packets (messages). The WEB browser application 101 requests WEB pages from other locations on the network to browse (display) what is available at these locations. This process is known as "navigating" to sites on the WEB network. In particular, when the WEB browser application 101 "navigates" to a new location, it requests a new page from the new location (e.g., server computer system 103) by sending an HTTP-request message 104 using any well-known underlying communications wire protocol. HTTP-request message 104 follows the specific layout discussed above, which includes a header 105 and a URI field 106, which specifies the target network location for the request. When the server computer system machine specified by URI 106 (e.g., the server computer system 103) receives the HTTP-request message, it decomposes the message packet and processes the request. When appropriate, the server computer system constructs a return message packet to send to the source location that originated the message (e.g., the client computer system 102) in the form of an HTTP-response message 107. In addition to the standard features of an HTTP message, such as the header 108, the HTTP-response message 107 contains the requested WEB page 109. When the HTTP-response message 107 reaches the client computer system 102, the WEB browser application 101 extracts the WEB page 109 from the message, and parses and interprets the HTML code in the page (executes the WEB page) in order to display the document on a display screen of the client computer system 102 in accordance with the HTML tags.

This portion of Krishna appears in the Background section of the patent and describes nothing more nor less than the use of HTTP in Internet Protocol in the standard request/serve. It describes in the most basic terms how a client machine requests a Web page and a server serves the Web page in response thereto. It does not, nor is it intended to, describe anything but the basic well-known request/response protocol between a client and server on the Internet for serving Web pages.

This passage contains no mention of supplemental files within a Web page, let alone any mechanism for indicating the order in which such supplemental file are to be served. This passage does not contain any discussion of the order of anything, in fact.

Discussion

In response to the previous Office Action (issued in response to Applicant's Appeal Brief), Applicant explained how Anders, like the present invention, teaches a technique for indicating an order to serve supplemental files within a Web page, that technique being different than the conventional technique based simply on where the reference appeared in the Web page in the standard reading of the code from left to right and top to bottom. However, Anders' technique is entirely different that that of the present invention. Applicant also explained that Krishnan is entirely irrelevant and does not teach anything about the order in which supplemental files within a Web page are to be served. The cited portion of Krishnan does not even mention supplemental files.

In its response to the previous Office Action, Applicant requested clarification of the Action, noting that (1) the rejection contained numerous clerical, typographical and/or grammatical errors that made it very difficult to understand, and (2) provided no explanation of how the Office proposed the references to be combined.

Unfortunately, the present rejection appears to be a verbatim copy of the previous rejection, including all of the previously noted typographical, clerical, and/or grammatical errors.

In the Response to Arguments section of the final Office Action, the Office added that "Applicants argue that Anders does not teach constructing a queue in memory comprising a list of supplemental files". In response, the Office argued that "Anders does teach constructing a queue in a memory comprising a list of supplemental files as shown in Col. 11, lines 7-30".

This was only one of many arguments Applicant made and was, in fact, a peripheral argument. Rather than address this issue (which Applicant still disputes, but believes to be much less significant than all other arguments asserted in the previous response), Applicant will herein focus the Examiner's attention on more fundamental distinctions between Anders and the present invention.

Applicant's previous arguments might be summarized as follows:

(1) Anders teaches an entirely different technique than the present invention for setting the order in which supplemental files within a Web page are to be served;

(2) Anders teaches nothing that resembles the sequence number attribute embedded within the tag referencing the supplemental file;

(3) in Anders, the order of serving of supplemental files is dictated by a program external to the Web page, whereas, in the present invention, the order data is provided within the Web page itself;

(4) contrary to the Office's concession that Anders does not teach an order, Anders does teach an order (albeit based on an ordering technique totally different than that of the present invention), whereas Krishnan (which the Office cited for teaching an order) actually does not teach any order (it does not even mention supplemental files in a Web page);

(5) the Office failed to recognize that its assertion that Anders teaches that the Streaming Configurator parses the Web page to identify references to objects and their locations within the page, but that the designer supplies the display sequence information is not what is recited in claim 1, but rather is the opposite of what it recited in claim 1;

(6) Krishnan does not teach that for which it has been cited because it does not disclose a Web page containing order data indicating the order in which supplemental files are to be disclosed;

(7) even if Krishnan did teach such order data, the offered motivation for the proposed combination of features is improper;

(8) the Office did not set forth a prima facie obviousness case because it did not describe the proposed combination of the two references; and

(9) the Office did not set forth a comprehensible motivation for the proposed combination:

It is unclear why the Office has not addressed any of these arguments made by Applicant.

It seems that the Office is misunderstanding the claims. The rejection of claim 7, perhaps, most starkly reveals this misunderstanding. Claim 7 is a rather specific claim that recites that "said order data comprises attributes of [HTML] tags". Anders teaches nothing even remotely resembling that the order in which supplemental files are downloaded is dictated by attributes within the HTML tags that reference the supplemental files.

In rejecting claim 7, the Office asserted that Anders teaches wherein said code defining said Web page comprises HTML code, said references to supplemental files comprises HTML tags (col. 11, line 7 – col. 12, line 44).

This rejection misses the point of claim 7. Applicant freely concedes that it is conventional that the code defining the Web page comprises HTML code, and that the references to supplemental files comprise HTML tags. These recitations appear in the claim simply to set up the next recitation, i.e., that the order data is disclosed in the attributes of those tags. Thus, the Examiner failed to mention the most significant part of claim 7 in the rejection. Anders does not teach this.

Anders dictates the order in which the supplemental files are downloaded in a completely different manner. Specifically, Anders' server builds the data stream 190 using a software module that Anders calls the Publisher 210 (see Figure 11). In Anders, the user specifies the order in which supplemental objects/files are downloaded by the browser, but the information dictating the order is not embedded within the main Web page itself. Rather, the order is determined by an external software module, namely, the Stream Configurator 212 in the Publisher 210. Thus, while Anders' technology does permit the server to dictate the order in which supplemental files are delivered to the browser, it does so in a way that is entirely different than what is claimed in the present application.

There is nothing in Anders like what is claimed in claim 7. Claim 7 recites an entirely different way to dictate the order of serving of supplemental files and, therefore, patentably distinguishes over the prior art

This patentable distinction appears in claim 1 also, albeit more broadly than claim 7. Referring to claim 1, Anders does not disclose (1) "parsing the code comprising the requested page to detect data within the code that indicates an order in which said supplemental files are to be served, said order data comprising data other than the order in which said supplemental files appear in said code defining said Web page". Anders obviously cannot meet this limitation because the order is given by an external program to the Web page, namely, the aforementioned Stream Configurator.

Independent claim 9 also distinguishes over Anders. Claim 9 includes the limitation of "second code indicating an order in which said supplemental files are to be rendered, said second code associated with each of said references and comprising an attribute of a tag associated with said supplemental file". This is similar to the recitation in claim 7. Hence, claim 9 distinguishes over Anders and Krishnan for reasons similar to those discussed above in connection with claim 7.

Independent claim 12 also distinguishes over Anders by virtue of reciting "program code for parsing said code defining the Web page to detect said order data". Since, as previously discussed, the sequence information is not in the Web page in Anders, it obviously cannot retrieve that information by parsing the code of the Web page.

Anders does not obtain the order data from inside the web page. Accordingly, independent claim 12 patentably distinguishes over Anders.

Independent claim 19 includes the limitation "code for parsing said code defining a Web page to detect said order data". In Anders, as previously mentioned, the display order is not found in the Web page and, therefore, this limitation is not met.

All of the dependent claims distinguish over the prior art for at least the reasons set forth above with respect to the independent claims from which they depend. However, the dependent claims also add even further patentably distinguishing recitations. See, for instance, claims 8, 11, 17, 18, 21, and 22 which add further

recitations about the order data being found in the attributes of HTML tags within the Web page.

In view of the foregoing amendments and remarks, this application is now in condition for allowance. Applicant respectfully requests the Examiner to issue a Notice of Allowance at the earliest possible date. The Examiner is invited to contact Applicant's undersigned counsel by telephone call in order to further the prosecution of this case in any way.

Respectfully submitted,

Dated: August 8, 2006

/Theodore Naccarella/

Theodore Naccarella
Registration No. 33,023
Synnestvedt & Lechner LLP
2600 Aramark Tower
1101 Market Street
Philadelphia, PA 19107
Telephone: (215) 923-4466
Facsimile: (215) 923-2189

Attorney for Applicant

TXN:pmf

S:\I\IBM\IBM Raleigh RSW\Patents\P24723 USA\PTO\Response to Final OX 3.doc